9- Write a MATLAB code to design an ANFIS model to approximate the following function:

$$f(x) = -2x - x^2, \text{ where } x \in [-10 , 10].$$

Show the best values of the optimized parameters after leaning or training is done.
To build the ANFIS model, consider the following Specs:

- **Use 60 training data pairs of input and output.**
- **Use 3 MFs of triangular type for the input *x*.**
- **Use the BP algorithm as a learning (or optimization) algorithm.**
- **The no. of iterations (or epochs) in the training process = 100**.
    - The total No. of rules = 3 rule
    - The triangular MF has 3 control parameters.
    - No. of premise parameters = No. of control parameters for MF x total No. of MFs

      = 3 x 3 = 9 parameter
    - No. of consequent parameters = (No. of inputs + 1) x No. of rules

      = (1 + 1) x 3 = 6 parameter
    - The total No. of ANFIS parameters

      = No. of premise parameters + No. of consequent parameters

      = 9 + 6 = 15 parameter

```
% First, we randomly obtain 60 training data pairs of input and output
% use the following form of random distribution
```

% $x_i = x_i^L + (x_i^U - x_i^L)$.*rand (No. of required points , 1);

```
>> x1 = -10 + (10+10).*rand(60,1); % x1 is a column vector with dimension 60x1
>>  y = -2.*x1- x1.*x1;          % y is a column vector with dimension 60x1
>> x = [x1  y];             % the training data set
>> n = 60;                    % no. of samples of the training data set
>> numEpochs = 100;
```

```
>> numMFs = 3;
>> mfType = ' trimf ';     % we can use 'gaussmf ' or ' trapmf ' or 'gbellmf '
                                   % build ANFIS model with the default initialization of parameters
                                     % Save initial ANFIS model as fismat1
>> fismat1= genfis1(x,numMFs,mfType);
                                  % Or we can use: fismat1=genfis1(x,[3 5],char('trimf ','gaussmf '));
                   % Train the initial ANFIS model over the training data set x
                                % Save the trained ANFIS model as fismat2
>> [fismat2,trn_mse,tst_mse] = anfis(x,fismat1,numEpochs,NaN,x,0);
  % NaN represent the default values of training options like training epoch number,
  % training error goal,initial step size, step size decrease rate and step size increase rate
  % trn_mse is avector contains the values of mean square errors during the epochs of
training process
 % tst_mse is avector contains the values of mean square errors during testing process
 % 0 is used for the BP  learning algorithm and 1 for the HL learning algorithm
 >> trn_out  = evalfis(x(:,1),fismat2);     % the values of outputs for the trained ANFIS
 >> [x(:,2)  trn_out]         % view the desired output and the trained ANFIS output
 >> AMSE = mean(trn_mse);    % the average of mean square errors over the 100 epochs.
 >> epoch = 1:numEpochs;
 >> plot(epoch, trn_mse)       % curve for mean square error vs. epochs
 >> yy = [1 : n ];                     % Data Set Index
 >> plot(yy,x(:,2),'x',yy,trn_out,'o')
```

10- In problem 2, state which of the following steps improve the approximation results for ANFIS:

- **Change the No. of MFs for the input *x* to 5.**
- **Change the type of MFs to Gaussian.**
- **Chang the learning algorithm to HL.**
  - The total No. of rules = 5 rule
  - The Gaussian MF has 2 control parameters.
  - No. of premise parameters = No. of control parameters for MF x total No. of MFs

= 2 x 5 = 10 parameter

- No. of consequent parameters = (No. of inputs + 1) x No. of rules

  = (1 + 1) x 5 = 10 parameter

- The total No. of ANFIS parameters

  = No. of premise parameters + No. of consequent parameters

  = 10 + 10 = 20 parameter

```
>> x1= -10+(10+10).*rand(60,1);
>> y= -2.*x1-x1.*x1;
>> x=[x1 y];
>> n=60;
>> numEpochs=100;
>> numMFs=5;
>> mfType='gaussmf';
>> fismat1=genfis1(x,numMFs,mfType);
>> [fismat2,trn_mse,tst_mse]= anfis(x,fismat1,numEpochs,NaN,x,1);
>> trn_out= evalfis(x(:,1),fismat2);
>> [x(:,2) trn_out]
>> AMSE= mean(trn_mse);
>> epoch= 1:numEpochs;
>> plot(epoch,trn_mse)
>> yy=[1:n];
>> plot(yy,x(:,2),'x',yy,trn_out,'o')
```

11. Simulate the behavior of the system (the system output is *x*):

$$\ddot{x} = -0.6\,\dot{x} + f(x)$$

Where $f(x) = -2x - x^2$ is the same as in problem 8.
First, build your simulink model with the exact expression for $f(x)$ then, replace $f(x)$ by the ANFIS fuzzy model that you reached to in problem 8 and compare the results. To run the simulink model use the following initial values:

$$x(0) = -0.2 \text{ and } \dot{x}(0) = 0.5$$

REPORT